Mathematical models applied to on-chip network on FPGA for resource estimation

Virginie Fresse, Catherine Combes, Hatem Belhassen

Mathematical models applied to on-chip network on FPGA for resource estimation

Virginie Fresse, Catherine Combes, Hatem Belhassen

Hubert Curien Laboratory UMR CNRS 5516, Jean Monnet University, University of Lyon,
18 Rue du Professeur Benoît Lauras
42000 Saint-Etienne, FRANCE
e-mail: virginie.fresse@univ-st-etienne.fr, catherine.combes@univ-st-etienne.fr

Abstract— *One main challenge of prototyping a SoC (System on Chip) on FPGA (Field Programmable Gate Array) is to tune at best the communication architecture according to the task graph of an application and the available resources of the chosen FPGA. The exploration of the potential design candidates is time consuming, tedious and does not scale. The sheer number of parameters leads to a wide design space that cannot be explored in a limited time.*

The aim of this paper is to identify mathematical models applied to NoC to estimate FPGA resources. Mathematical models are obtained from a database containing a set of observed results. Using the database, the Pearson's correlation coefficient and the variable clustering are used to set the most appropriate variables and constants. The mathematical models are obtained and then validated with a set of experimental results. The validation shows that the error rate between observed results and the analytically estimated results is less than 5%. The designer can therefore tune the NoC in shorter exploration time.

Keywords — Design Space Exploration, NoC dimensioning, FPGA, mathematical modeling, resource estimation.

I. INTRODUCTION

Systems On Chip (SoC) using Network on Chip (NoC) are the most appropriate systems for real time embedded applications. The SoC is a set of hardware or software IPs (Intellectual Properties) connected to the NoC. NoCs are emerging communication structures as they provide high bandwidth and high scalability with low power.

In the design of SoC, the number and type of IPs are extracted from the algorithm. Signal and image processing algorithms are mainly described as a graph with functions and flow related to data or control dependencies. In most cases, one function is considered as an IP in the SoC. The flows between functions are implemented by means of the communication structure (NoC). According to the number of IPs, the communication structure is tuned. Tuning the NoC consists in selecting appropriate parameters of the NoC. The number of parameters and their associated values are high. Exploring all appropriate solutions is an intensive time process because of the sheer number of parameters required for the NoC. The designer selects the NoC without exploring all candidates, gaining significant time in the development process but the chosen solution is not always optimal.

The Field Programmable Gate Array (FPGA) devices are widely used for prototyping systems. FPGA can be used to emulate the NoC performances within a fast design space exploration cycle. The performance metrics on FPGA are timing (bandwidth, latencies), area and energy consumption. Emulation gives precise timing and power evaluations in a shorter cycle compared to simulation [1]. Emulation on FPGA is not suitable for area estimation. Each NoC candidate has to be synthesized then placed and routed to obtain the number of resources required. Area estimation can be explored after either the synthesis or the place and route process. These processes are time consuming. With large size of FPGA, this process can run for several hours for one NoC candidate. Therefore it is not possible to synthesize all candidates to evaluate the number of resources required.

Area estimation is important in order to find architectural solutions that: 1) suit to the target FPGA 2) correspond to application requirements 3) provide efficient timing results. The aim of this work is to provide a methodological framework based on mathematical modeling. The identified models help the designer to select the appropriate NoC candidates from a restricted number of implementation with FPGA and algorithm constraints.

The contribution of this paper is to show the feasibility of creating a mathematical model for sizing a NoC on FPGA. This model was validated by a comprehensive set of experimentations. It guarantees a reduction of time necessary to the design space exploration (DSE) of NoC in the field of signal and image processing.

The paper is organized into 9 sections. Related works are given in section 2. The design space exploration of NoC on FPGA including basis elements (NoC, FPGA, application and devices) is detailed in section 3. The methodological framework is described in section 4. The development of the mathematical models is presented in section 5. Validation of both experiments XP1 and XP2 for different sizes of NoC on a set of FPGAs are described in section 6. The impact of the synthesis optimization goals is studied in section7. The limitations of the models are in section8. Conclusion and future works are in section 9.

II. RELATED WORKS

NoCs have emerged as efficient scalable and low power communication structures for many-core SoC (System On Chip including several hundred or thousands of cores). Many NoCs are designed for FPGA devices [2][3][4] and application-specific NoC design flows are proposed [13]. In the design flow, the application described as a task graph is mapped to the topology graph. The topology graph is the NoC

structure with all parameters already specified. These design flows do not explore the design space of the NoC.
Design space explorations for the NoC are mainly based on power consumption and timing [1][5][6][7][10]. ORION is a tool designed for fast and accurate power and area model on Integrated Circuit (IC) [7]. The tool explores the area occupied (mm2) and the power (mW) by the NoC used on 65 nm chips when routers and links increase. The 3D Tezzaron design flow also explores the 3D NoC to reduce the area of the chip and interconnects on ASIC to optimize power [10]. A system level approach is proposed to explore the NoC design space with an objective to minimize the energy consumption and link bandwidth (timing). These works concern power and timing evaluation for ASIC.
NOCDEX is a tool to evaluate the impact of various options on area, number of cycles and execution time on FPGA [8]. The tool evaluated the number of cycles according to the number of slices and the maximum frequency for a cascade NoC with 4 masters and 4 slaves.
Power models at different abstraction levels have also been proposed for a variety of networks in the past. Models for resource estimation of the NoC on FPGA have not been fully explored yet. A power area analysis of NoCs in FPGAs has been proposed in [9]. The analysis is based on the analysis of power and area of the router for the 4×4 torus topology. This work only considers the routing blocks, no any others blocks or routing. The number of links varies according to the position of the router so that it has a huge impact of the total number of resources. It is necessary to analyze the global structure to obtain a precise model depending on the topology. We propose to explore the design space of NoCs to estimate the area metric on FPGA. Explorations are constrained by the target FPGA and the data flow graph from the application.

III. DESIGN SPACE EXPLORATION OF NOC ON FPGA

Design Space Exploration (DSE) refers to the activity of exploring design alternatives prior to implementation [12].
The challenge of DSE is to explore the sheer size of the design space and to find the best candidates. Typically a large system has billions of possibilities as parameters of NoCs are abundant. The designer must select the topology, the number of nodes, the size of flits, the commutation mode, the routing algorithm, the size of buffers and many other parameters. Enumerating every point of the design space is prohibitive and is time consuming [11].
Moreover, the exploration of NoC candidates on FPGA is huge as the number of FPGAs is high. The portfolio of Xilinx company includes six FPGA families (Spartan, Artix, Kintex, Kintex Ultrascale, Virtex and Virtex ultrascale) [15]. Each family has around hundreds of devices.
The estimation of resources for each NoC candidate on each FPGA device is long. Observed resources are obtained from a synthesis process whose time depends on the size of the NoC (29 minutes for a medium size and several hours for a large size). Exploring all points on the design space can take several weeks.

The methodological framework proposed is based on the following elements described below:

- The NoC structure,
- The FPGA devices.

A. Network On Chip (NoC)

NoC are communication structures proposed as a solution for the communication challenge. NoC architecture is composed of several basic elements depicted in Fig. 1 :

- Network Interface (NI): it enables PE (Processing Element) to communicate with routing node they are connected to.
- Routing node (or switch): according to the routing algorithm, the switch sends packets to the appropriate link in the network.
- Links: connect the routing switches together or switches to NI.
- Processing Element: these units correspond to various modules of SoC, such as IP blocks, memories, processors.

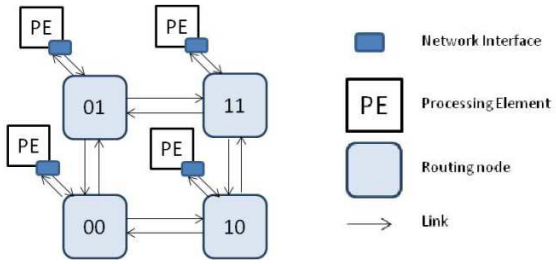

Fig 1. Basic elements for the NoC structure.

Many topologies can be considered for NoC structures: mesh (a), torus, ring (b), tree (c)…. The mesh topology is the most appropriate for FPGA devices (as depicted in Fig. 1). The following work uses mesh-based NoC but the framework can be extended to others.
One PE sends messages to another PE through the NoC. Messages are compacted and divided into packets, which are divided into parts of the size of a flit. Flits (Flow Control Units) can be classified based on their position inside the packets as header, tail or payload.

B. FPGA

An FPGA (Field Programmable Gate Array) is a programmable logic device used in various applications requiring rapid prototyping of digital electronics (telecommunication, image processing...). Modern FPGAs are now able to host processors cores as well as several IP blocks to perform efficient prototyping of embedded systems.
However, the designer can obtain, after the synthesis process, a prevision of FPGA resources. Resources are LUT (Look Up Table), MLUT (Memory LUT), FF (Flip Flop), Buffers and I/O (Input and Ouput). The resource allocation is given in the post-synthesis report. This allowance depends on CLB (Configurable Logic Block) that define the internal architecture of the FPGA.

IV. METHODOLOGICAL FRAMEWORK

A. Description

The objective is to mathematically model the relation between the input configuration of the NoC and material resources used without going through the step of synthesis. So we have to identify links between NoC input variables and the FPGA resources used for the NoC (LUT, MLUT, FF).
The variables considered in the mathematical models are:

- n_1: the number of routers in the X-axis.
- n_2: the number of routers in the Y-axis.
- n_3: the depth of buffer.
- n_4: the size of flit.

Other inputs of the NoC are set (i.e. are constants) such as routing algorithm, flow control and the number of virtual channels (for the use of the credit based control flow).

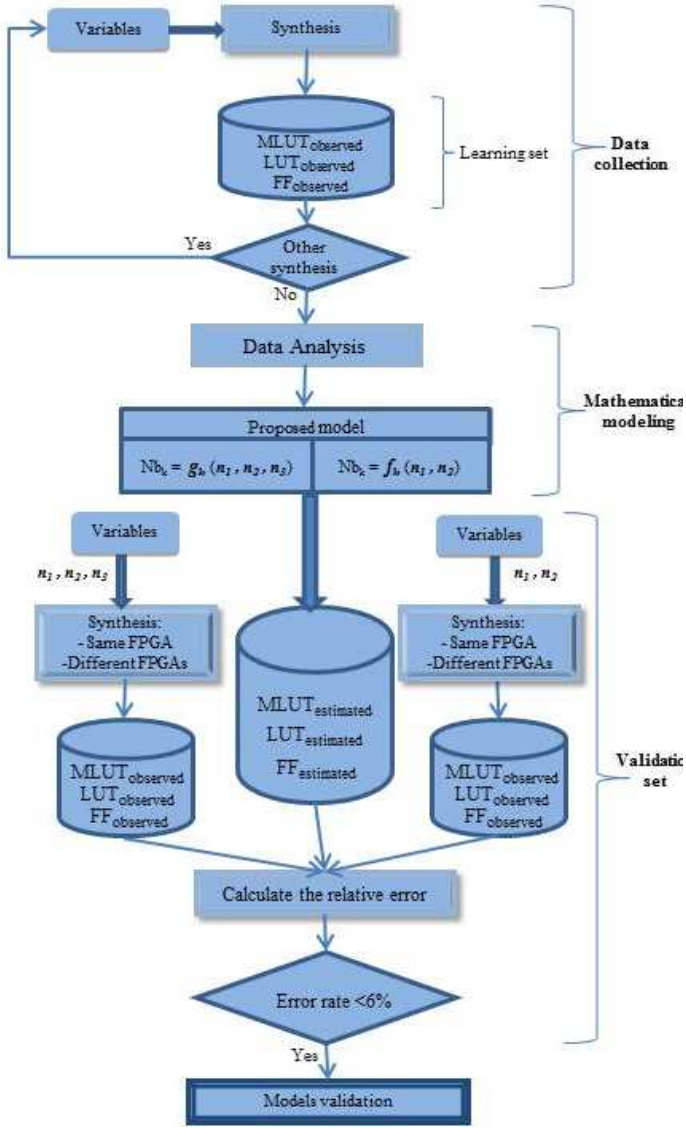

Fig 2. Methodological framework

The methodological framework is proposed in Fig 2. The results obtained are the number of LUT, MLUT and FF extracted from each synthesis process. They are stored in a database. The number of synthesis should be enough to get a full set of observed results. As long as the database is not complete, synthesis of the NoC according to variables are repeated.
Then, from the complete database of observed results, data analysis is done to obtain links between variables and LUT, MLUT, FF. In the first experiment, variables are (n_1, n_2). In the second experiment, variables are (n_1, n_2, n_4).
In the first and second experiment, once the learning set is done, several steps are involved. Starts by analyzing data, deducing mathematical models, estimating resources, synthesizing additional NoC configurations on the same FPGA used on the learning set or synthesizing existing data base Noc sizes on different FPGAs, then calculating the relative error between observed and estimated resources, at the end, if the error rate is less than 6%, models are validated.

B. Context of XP1

The NoC used in the following experiments is the NoC Hermes. It was developed by the Catholic University of Rio Grande do Sul, in Porto Alegre, Brazil [2]. This NoC is based on a 2D Mesh switch. The main components of this infrastructure are the Hermes switch and IP cores (Fig 3.). The Hermes switch has routing control logic and five bi-directional ports. All ports contain input buffers for provisional storage of information.

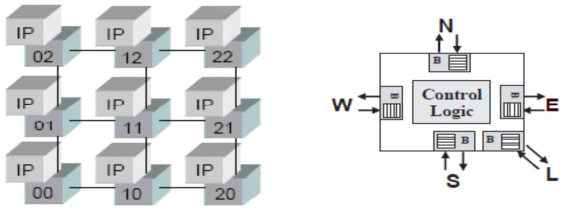

Fig 3. The Hermes NoC and its switch architecture.

The platform is the VC707 Evaluation Platform with a Virtex 7 XC7 VX485 FPGA (containing 303,600 LUTs, 130,800 MLUTs, and 607,200 FF, registers and IO…).
The experiments are carried out within the Atlas tool and then VIVADO 2012.3 as depicted in Fig 4. The synthesis tool is the XST tool (Xilinx Synthesis Tool).

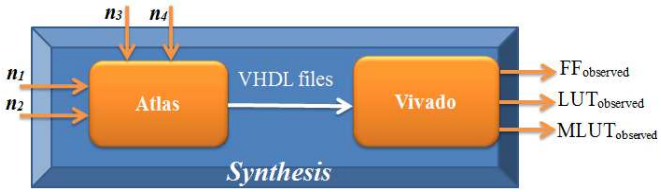

Fig 4. NoC synthesis for XP1

The objective is to find $f_k(n_1 , n_2)$ which corresponds to the identification of Nb_k (number of resources k) for input constants (n_3: buffer depth {32} and n_4: size of flit {16}) and input variables (n_1: number of routers in the X-axis {3..16} and n_2: number of routers in the Y-axis {3..8}).

C. *Context of XP2*

The NoC used in the following experiments is also the NoC Hermes. The objective of this second experiment is to find g_k (n_1, n_2, n_4) which corresponds to the identification of Nb_k (number of resources k) for input constants (n_3: buffer depth {32} and input variables (n_1: number of routers in the X-axis {3..16} and n_2: number of routers in the Y-axis {3..8}and n_4: size of flit {16, 32, 64}). The context of the second experiment is depicted in Fig 5.

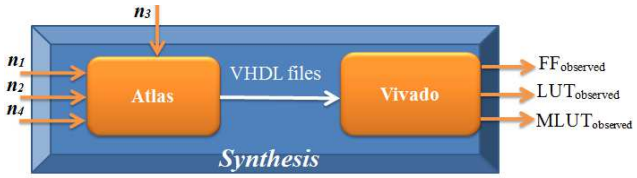

Fig 5. NoC synthesis for XP2

V. MATHEMATICAL MODELING

Before defining the mathematical models, the analysis of data is achieved to define the most appropriate variables and the order of resources.

A. *Data analysis*

The aim is to identify from the data analysis links between input variables (n_1, n_2, n_4) and FPGA resources LUT, MLUT and FF. The variable n_3 is not considered as the buffer depth remains always identical. This is due to the different kind of memory blocks that are used according to the NoC: the number does not change, only the type (i.e. the size of the block). The Pearson's correlation coefficient is first observed to measure the strength of a linear association between two variables (Table 3).

Table 3. Pearson's correlation coefficient.

	n_4	n_2	n_1	*FF*	*LUT*	*MLUT*
n_2	0.17					
n_1	-0.157	-0.0237				
FF	0.364	0.509	0.531			
LUT	0.460	0.503	0.467	0.992		
MLUT	0.627	0.463	0.339	0.932	0.971	
$n_1 \times n_2$	-0.056	0.462	0.68	0.88	0.821	0.643

There is a strong correlation between the observed resources.

While analyzing similarities between groups, the strongest correlation is between FFs and LUTs (0.992). There are also strong correlations between MLUT and LUT, and also FF and MLUT.

There is also a strong correlation (lesser but significant) between $n_1 \times n_2$ and the number of FF and LUT.

One unexpected correlation is the correlation (in blue) between $n_1 \times n_2$ and the FPGA resources: 0.64 with MLUT, 0.88 with FF and 0.882 with LUT. This correlation is higher than the correlations between the FPGA resources and n_1 or n_2. It is also observed that n_1 has a higher correlation with FPGA resources than n_4 and n_2. The impact of the size of flits is higher than the number of nodes in *Y*, but lower than the number of nodes in *X*.

But it is also possible to cluster variables in terms of their correlations. Two variables have a pair of values for each sample, and measures of distance and dissimilarity between these two column vectors can be considered. The similarity between variables is measured: this can be in the form of correlation coefficients or other measures of association. The result of a cluster analysis is a binary tree, or dendrogram, with *n-1* nodes. The branches of this tree are cut at a level of similarities obtained in our case by using correlation.

A strong correlation indicates a high degree of similarity. A weak correlation indicates a low degree of similarity. Similarities are depicted in Table 4 and the corresponding dendrogram is presented in Figure 6.

Table 4. Table of similarities in the melting step

Group number	Number of variables	Similarities between the group of variables
1	6	99.58
2	5	98.53
3	4	94.00
4	3	84.00
5	2	81.35
6	1	75.46

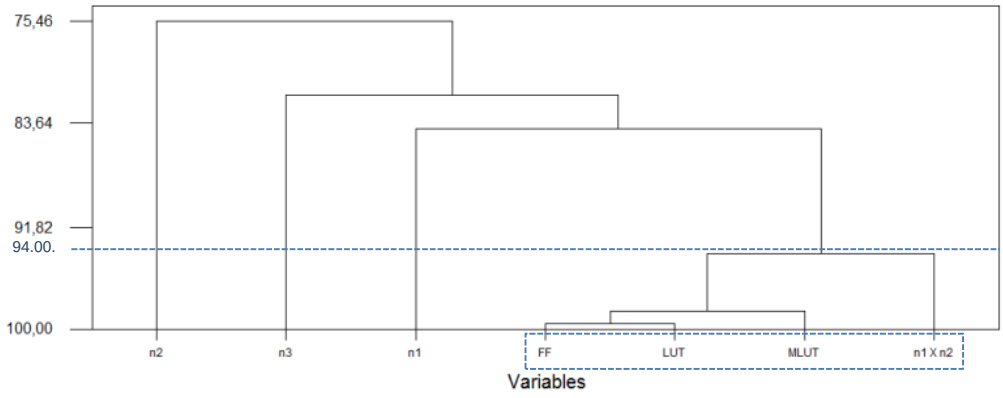

Fig 6. Similarities between the variables.

They are illustrated in the table of similarities in the melting step (Table 4). 6 groups are extracted. The first group (with a 99.58 similarity) has 6 variables. The second group is extracted for a similarity degree above 98.53. This group contains 5 variables. It indicates that two variables, FF and LUT, have a small distance. The last group contains one variable only.
The FPGA resources (LUT, FF and MLUT) are strongly correlated to ($n_1 \times n_2$) as the similarity degree is 94 (cf Table 4, group number 3).

So, models are carried out with the variable called $n_1 \times n_2$ instead of considering two independent variables n_1 and n_2. The mathematical models are first based on the number of MLUTs as the correlation is lower. The mathematical models are also studied from $n_1 \times n_2$ as this variable has the highest degree of similarities compared to n_1 only or n_2 only.

B. Mathematical models for XP1

As the number of routers is n_1 multiplied by n_2, the relation between $MLUT_{observed}$ and (n_1, n_2) must be considered. So dividing the number of $MLUT_{observed}$ per ($n_1 \times n_2$) is necessary to find those links between the number of MLUTs and one router. The phenomenon is illustrated in Fig 7. This figure shows 14 classes of measures corresponding to the 14 variations of n_1. And from a class to another there is a translation on horizontal and vertical axis.

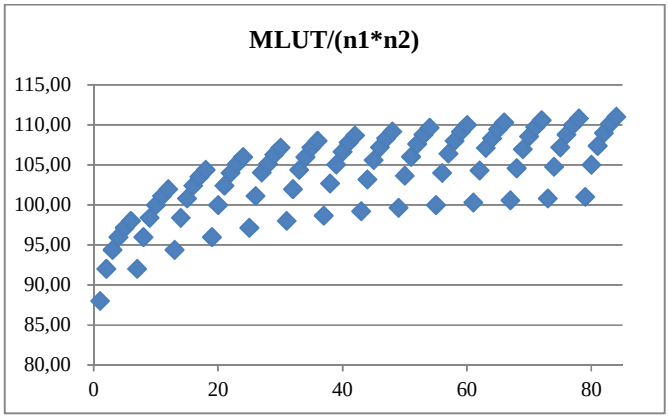

Fig 7. Number of MLUTs divided per ($n_1 \times n_2$)

Approaching each class by logarithmic trendlines (a logarithmic trendline is the best-fit curved line) confirms that $\boldsymbol{MLUT_{observed}/(n_1 \times n_2)}$ can be written as: ($\boldsymbol{a \times LN(n_2\text{-}2) + b}$) and there is provided the values of ***a*** and ***b*** in Fig 7.

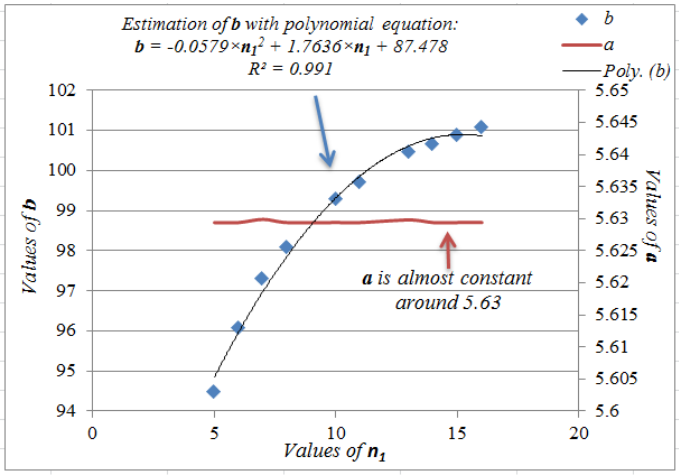

Fig 8. Variation of coefficients ***a*** and ***b***

Table 5. Coefficients to express FF, LUT according to MLUT

	Coefficient FF/MLUT	Coefficient LUT/MLUT	Coefficient LUT/FF
Average	3.9632	12.2779	0.32281
Median	3.9626	12.3027	0.32184
Standard deviation	0.0049	0.1243	0.00363
Min	3.9442	11.9239	0.31789
Max	3.9737	12.4518	0.33327
(Min+ Max)/2	3.95899	12.1878	0.32558
Coefficient chosen	4	12	1/3

The curves show that ***a*** is almost a constant value around 5.63 and ***b*** varies. Approaching ***b*** by a polynomial trendline allows us to deduce the corresponding formula:

$$\boldsymbol{f_{MLUT}(n_1, n_2) = (n_1 \times n_2) \times [5.63 \times LN(n_2 - 2) + (-0.0579 \times n_1^2 + 1.7636 \times n_1 + 87.478)]} \quad (1)$$

The strong correlation between MLUT and both LUT and FF is extracted in the previous section (Data Analysis). Therefore a coefficient to express LUT and FF according to $\boldsymbol{f_{MLUT}}$ can be found (table 5). With the chosen coefficients, $\boldsymbol{f_{LUT}}$ and $\mathbf{f_{FF}}$ become:

$$\boldsymbol{f_{LUT}(n_1, n_2) = 12 \times f_{MLUT}(n_1, n_2)} \quad (2)$$

$$\boldsymbol{f_{FF}(n_1, n_2) = 4 \times f_{MLUT}(n_1, n_2)} \quad (3)$$

Regarding to $f_{MLUT}(n_1, n_2)$ given in equation (1).

C. Mathematical models for XP2

The purpose of this second experiment is to express $\boldsymbol{g_k(n_1, n_2, n_4) = coeff_{kij} * f_k(n_1, n_2)}$. The size of flits is changed from 16 to 32 and the impact on the number of resources is analyzed. The number of resources changes according to the type of resources. There is one specific coefficient for MLUT (the value is 2), for LUT (the value is 1.477) and for FF (the value is 1.833) in this case.

Table 6. Coefficient *k* for flit from size 16 to 32

k = MLUT	*k* = LUT	*k* = FF
i= 16	*i*= 16	*i*= 16
j=32	*j*= 32	*j*= 32
$Coeff_{MLUT\ 16\ 32}$ = 2	$Coeff_{LUT\ 16\ 32}$ = 1.477	$Coeff_{FF\ 16\ 32}$ = 1.833

Then, for n_4=16 ->32

$$g_{MLUT}(n_1, n_2, n_4) = 2*f_{MLUT}(n_1, n_2) \quad (4)$$

and

$$g_{LUT}(n_1, n_2, n_4) = 1.477*f_{LUT}(n_1, n_2) \quad (5)$$

and

$$g_{FF}(n_1, n_2, n_4) = 1.833*f_{FF}(n_1, n_2) \quad (6)$$

Table7 and table 8 illustrate different coefficients while varying the size of flits (16, 32 and 64).

Table 7. Coefficient *k* regarding the size of flits (from *i* to *j*).

k = MLUT	*k* = LUT	*k* = FF
i= 16	*i*= 16	*i*= 16
j=64	*j*=64	*j*= 64
$Coeff_{MLUT\ 16\ 64}$ = 3.66	$Coeff_{LUT\ 16\ 64}$ = 2.437	$Coeff_{FF16\ 64}$ = 2.01

Table 8. Coefficient *k* regarding the size of flits (from *i* to *j*).

k = MLUT	*k* = LUT	*k* = FF
i= 32	*i*= 32	*i*= 32
j=64	*j*=64	*j*= 64
$Coeff_{MLUT\ 32\ 64}$ = 1.88	$Coeff_{LUT\ 32\ 64}$ = 1.64	$Coeff_{FF\ 32\ 64}$ = 1.05

According to the configuration of the size of flits from *i* to *j*, all coefficients called $Coeff_{kij}$ are different (see table 7 and table 8). The coefficient for MLUT, called $Coeff_{MLUTij,}$ is respectively 2 and 1.88 for flits (*i*=16 to *j*=32) and for flits (*i*=32 to *j*=64). This phenomenon is identical for *k*=*FF* and *k*=*LUT*.

VI. VALIDATION

The validation is first done on results extracted from FPGA already used. Then other implementations with random sizes of NoCs and different size of flits are done to validate the models.

A. *Validation of XP1 with different sizes of NoC on a Virtex7 FPGA*

To validate our models, random NoC architectures are selected. Varying the number of routers (n_1, n_2), synthesizing and comparing with analytically estimated results gives the error rates illustrated in Fig 9.

There are three classes of error rates. The first class is for n_1 = 3, and n_2 = {3..8}. The error rate is 4.20% for the number of MLUTs, 4.33% for FFs and 2.51% for LUTs. The maximum error rate for the first class is less than 5%. The second class is for n_1 = 4 and n_2 = {3..8}. The error rate for the number of MLUTs is 1.42%, 1.49% for FFs and 0.46% for LUTs. The maximum error rate for the second class is less than 2%. The third and last class is for n_1 = {5..16} and n_2 = {3..8}. The error rate for the number of MLUTs is between [-0.65%, 0.47%], for the number of FFs it is between [-0.83%, 0.45%] and for LUTs it is between [-0.66%, 1.63%].

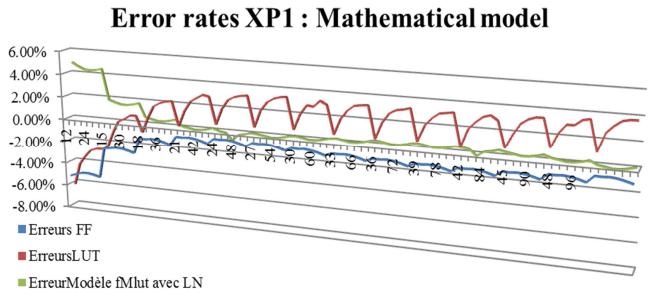

Fig 9. Error rates corresponding to f_k for ***k*** = FF, LUT, MLUT

Therefore, it is possible to estimate the number of resources for small sizes of NoC with an error rate less than ±5% (first class). The error rates for the third class are less than ±2%. This third class concerns large NoC having an important synthesis time. These error rates indicate that the number of estimated resources is near to post synthesis results.

B. *Validation of XP2 with different sizes of NoC on a Virtex7 FPGA*

The mathematical model validation is then done on previously unused FPGAs. The error rate between observed results and estimated is analyzed and presented in Fig 14. and Fig 11.

The error rate for a size of flit from 16 to 32 is around -6% for small size of NoCs. This error rate decrease to less than 4% for bigger sizes of NoC. This indicates that the analytically estimated results are a little bit bigger than the results obtained after synthesis (synthesized results) for small sizes of NoC. The analytically estimated results are smaller than synthesized results for bigger sizes of NoCs. For any cases, the error rate is low and the analytically estimated results are close to synthesized results. This phenomenon is identical when the size of flits changes from 16 to 64. The error rate is a little bit higher for smaller size of NoC (around -7%) and bigger size of NoC. This error rate remains fairly good.

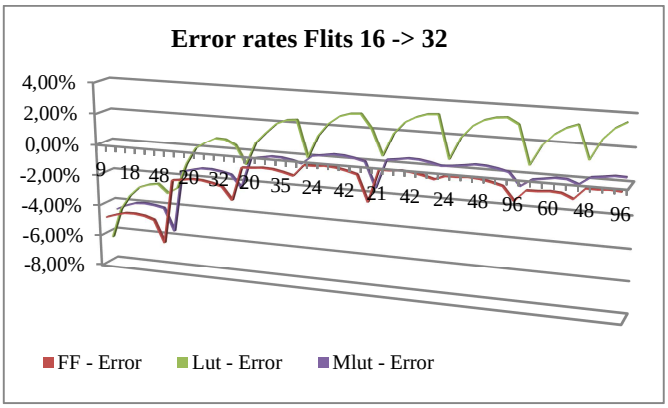

Fig 10. Rate error of XP2 model for f_k *when the size of flit* ***16-> 32***

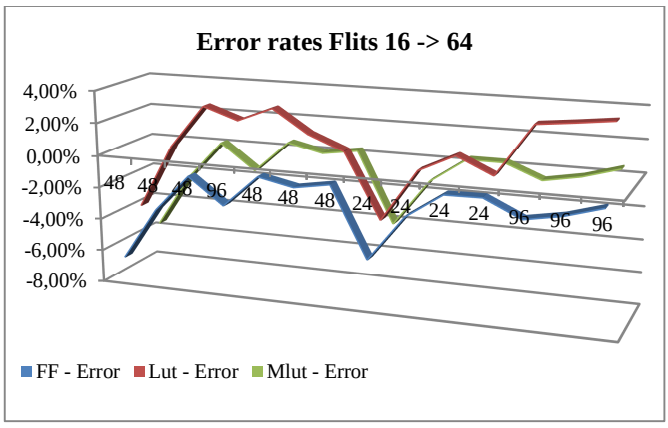

Fig 11. Rate error of XP2 model for f_k *when the size of flit* ***16-> 64***

C. *Validation using different FPGA devices*

To validate the experiments, three architectures of NoC varying the number of switches (n_1; n_2) were selected by drawing lots. Their sizes are {(7; 4), (11; 12), (5; 11)}. Then 20 FPGA from the 326 FPGA available in Vivado 2012.3 were selected randomly. The FPGA available on Vivado are distributed as follows {194 Virtex7, 92 Kintex 7 and 40 Artix7}.

After synthesizing on the 20 different FPGAs, the number of resources is the same for each size of NoC. Fig. 12 depicts that the error rates vary from -0.5% to 1.2%. This result suggests that the mathematical models can correctly estimate the numbers of MLUTs, LUTs and FFs for XILINX families.

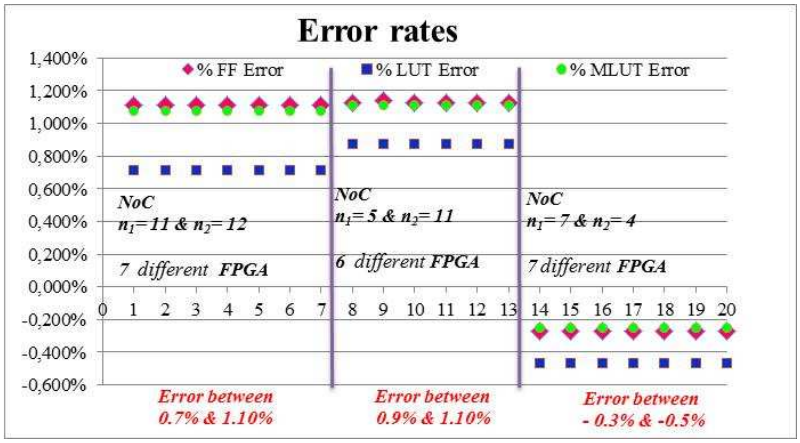

Fig 12. Error rate of the model for f_k *on different FPGA*

VII. IMPACT OF SYNTHESIS OPTIONS

The objective is to analyze the impact of the synthesis options. The options concern the area or speed optimization with three optimization efforts (normal, high and fast) for the Xilinx Synthesis Tool (XST). 6 combinations (Table 9) can be considered when synthesizing the NoC. The NoC used is the Hermes NoC. The routing algorithm is identical to the previous experiments (XY), the mesh topology too. The depth of buffer and the size of flit are 16. The main difference concerns the flow control. The previous flow control was credit based with 2 virtual channels. The flow control is now replaced by the handshake protocol.

The environment used for the experiments is Xilinx ISE 14.7. The synthesis is done with two sizes of NoC (7×7 and 14×14) on the Virtex 7 FPGA (VC707 evaluation platform).

Table 9. Synthesis options in XST (with ISE 14.7)

Combinations	Optimization Goal	Optimization Effort
1	Speed	Normal
2	Speed	High
3	Speed	Fast
4	Area	Normal
5	Area	High
6	Area	Fast

The synthesis times for the size 14×14 are depicted in Fig 14. The fastest time is obtained with the fast mode. The normal optimization goal ensures faster synthesis than the high optimization goal.

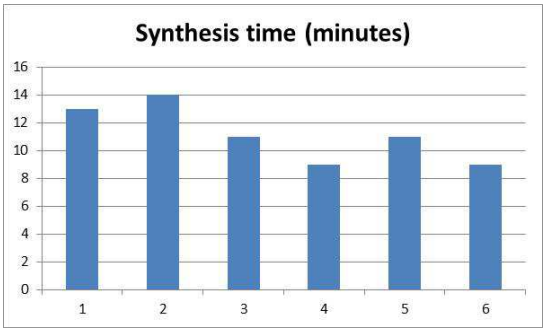

Fig 13. Synthesis times for the 6 combinations of the 14×14 Hermes NoC.

Table 10 and table 11 give the number of resources for respectively the area and resources optimizations.

Table 10. Number of resources according to the area optimization goal.

14x14	FF	38 170	38 170	38 184
7x7	FF	8 987	8 987	8 987
14x14	LUTs	109 383	109 383	109 737
7x7	LUTs	25 525	25 525	25 539
14x14	MLUTS	14 784	14 784	14 784
7x7	MLUTS	3 472	3 472	3 472

The optimization efforts have not impact on the number of FFs and MLUTs when optimizing the area. The normal and high efforts give the same number of FFs. The fast optimization effort modifies the number of LUTs. This can be considered insignificant as it only represents 0.32% of added LUTs for the 14×14 and 0.2% for the 7×7.

Table 11. Number of resources according to the speed optimization goal.

		1	2	3
14x14	FF	38 170	38 170	38 170
7x7	FF	8 986	8 986	8 986
14x14	LUTs	119 580	119 580	132 729
7x7	LUTs	27 816	27 816	34 080
14x14	MLUTS	14 784	14 784	14 784
7x7	MLUTS	3 472	3 472	3 472

For the speed optimization goal, the number of resources is identical for the normal and high options. The fast option changes the number of FFs and LUTs. In the fast mode, 22% and 11% added LUTs are required for the 7×7 and the 14×14.
Therefore, the number of MLUTs does not depend on the synthesis options. The number of FFs remains identical for 5 synthesis options. There are only 14 more FFs for the 14×14 NoC using the fast area option. This is not significant compared to the 38 710 FF used.
The only main difference lies in the number of LUTs used.
This analyze confirm the methodology proposed in section V. Mathematical models are proposed to first estimate the number of MLUTs (f_{MLUT}) according to the input variables. Then the number of FFs (f_{FF}) and LUTs (f_{LUT}) are extracted according to the number of MLUTs (f_{MLUT}).

To extract mathematical models, the optimization goal must be wisely selected. According to the previous results, the normal area mode should be selected. This mode gives the same result as the high optimization goal with a shorter synthesis time and it gives better results (fewer resources) than the fast optimization level.

As the resource results are similar for the normal and high optimization goals, the mathematical model can be defined from the normal mode using the area optimization.

VIII. LIMIT OF THE MODELS

The mathematical models are extracted from a specific NoC with variables and constants. In the previous experiments and the correlation analysis, 3 variables are extracted and other parameters are used as constants. The depth of buffer (variable n_3) became a constant as it has no impact on the number of resources. Other parameters were first considered as constant to restrict the number of variables.
In this section, 2 other variables are considered: the routing algorithm (n_5) and the flow control (n_6).
The experiments conducted on the Hermes NoC with different routing algorithms (semi adaptive and determinist routing algorithm) show that such algorithms do not have any impact on the number of resources (7 lines are overlapped). Fig. 15 depicts the number of FFs according to the routing algorithms and for different sizes of NoC. This behavior is identical for LUTs and MLUTs. The routing algorithm (n_5) is then considered as a constant when building the mathematical model. This is true for these kinds of routing algorithms. The use of more sophisticated algorithms can lead to more added resources. In this case, n_5 will be considered as a variable used to build the mathematical model).

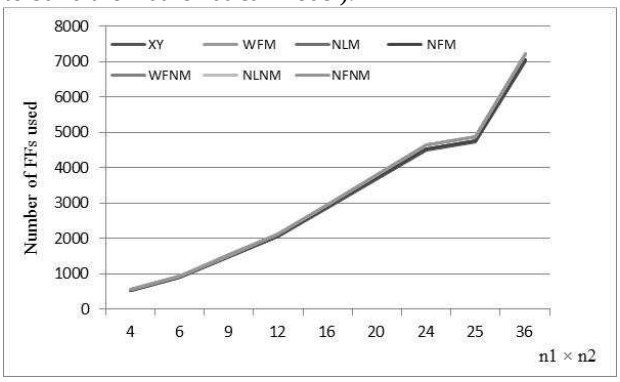

Fig 14. Number of FFs for different sizes of Hermes NoC using different routing algorithms.

The experiments are conducted for the flow control (n_6) using two types: the handshake and the credit based with 2 virtual channels, in Fig 15. The number of resources significantly changes according to the flow control.

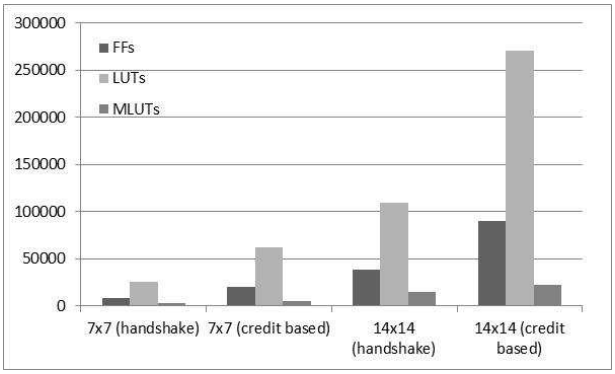

Fig 15. Number of resources according to the flow control.

A comparison between both flows is done in the table 10 (the number of added resources for the credit based compared to the handshake. The resources for the handshake are extracted from the mathematical models. The resources for the virtual channel are obtained from the synthesis of the NoC. The number of FFs is 2.3 more for the virtual channel than the handshake for both sizes of NoC, 2.44 more for the LUTs and 1.5 more for the MLUTS.

Table 10. Ratio of resources between both types of flow for two sizes of NoC (credit based/handshake).

	FFs	LUTs	MLUTs
7x7	2,313	2,443	1,497
14x14	2,358	2,469	1,522

The exploration of the NoC should also consider the control flow (variable n_6). Changing the flow control leads to define coefficients for each type of resource from the initial model as it has been done for n_4.

IX. CONCLUSION AND FUTURE WORK

In this paper, the feasibility to identify mathematical models for exploring the NoC on FPGA devices has been shown. The number of FPGA resources (LUT, MLUT and FF) can be estimated using these models and without experiments. The designer can explore the entire design space to find the most appropriate candidate in shorter time. The time saving is significant as the exploration with mathematical models takes only few minutes and with experiments takes few days. The designer can also explore all FPGA candidates without increasing the exploration time.

Mathematical models for a new NoC structure (topologies, flow control…) should be based on the analysis of the correlation of NoC variables and FPGA resources and also the correlation of Pearson 2-2. These analysis leads to select the input variables and the input constants. Input data can be one parameter or a combination of parameters. These analysis help the designer to order the Nb_k (number of resources k) for input constants and variables. The mathematical models obtained can estimate the number of resource with the lowest error rate.

Future work is to define the impact of the variable n_6 in the mathematical model and to analyze these models for other topologies.

ACKNOWLEDGMENT

Funding for this project was provided by a grant from *la region Rhone-Alpes*.

REFERENCES

[1] N. Genko, D. Atienza, G. De Micheli, J.M. Mendias, R. Hermida, F. Catthoor: "A Complete Network-On-Chip Emulation Framework". Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Page(s):246-251 Vol. 1, 2005.

[2] F. Moraes, N. Calazans, A. Mello: "HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip", Integration, the VLSI Journal, vol. 38, no. 1. pp 69–93. Oct. 2004.

[3] C. A. Zeferino, A. A. Susin: "SoCIN: A Parametric and Scalable Network-on-Chip", Proc. 16th Symposium On Integrated Circuits and System Designs, pp 169-174. 2003.

[4] C. Hilton, B. Nelson: "PNoC: a flexible circuit-switched NoC for FPGA-based Systems", in Field Programmable Logic, Aug. 2005. pp 24–26.

[5] A. Banerjee, R. Mullins, S. Moore "A power and energy exploration of Network on Chip architecture". Proceedings NoC, pp163-172, 2007.

[6] S. Bhat "Energy models for network on Chip Components". MS thesis, department of mathematics and computer science, Royal Institute of Technology, Eindoven, 2005.

[7] A.B. Kahng, B. Lin, L.S. Peh and K. Samadi "ORION 2.0: a fast and accurate NoC power and area model for early stage design space exploration". Design Automation and Test in Europe Conference and Exhibition, April 2009.

[8] X. Li, O Hammami "NOCDEX: Network on Chip Design Space Exploration Through Direct Execution and Options Selection Through Principal Component Analysis". International Symposium on Industrial Embedded Systems IES October 2006, France.

[9] M. Binesh Marvasti, T.H. Szymanski "A power-area analysis of NoCs in FPGAs". IEEE International of SoC Conference, 12-14 September 2012.

[10] R.K. Jena "System Level approach to NoC Design Space exploration". International Journal of Information and Electronics Engineering, vol 2, n° 2, march 2013.

[11] F. G. Villamon "Design Space Exploration for Networks on Chip". Phd report, University of Valencia, Spain, 2011.

[12] E. Kang, E. Jackson, W. Schulte "An approach for effective Design Space Exploration". FOCS'10 Proceedings of the 16the Monterey conference on Foundations of computer software: modeling, development and verification of adaptive systems, pp33-54, 2010.

[13] P. K. Sahu, A. Chattopadhyay "A survey on application mapping strategies for Network On Chip design". Journal of Systems architecture, Elsevier, 59, pp 60-76, 2013.

[14] Xilinx Inc
http://www.xilinx.com/support/documentation/user_guides/ug474(V1.5)_7Series_CLB.pd
Xilinx Inc " 7 series FPGA overview"

[15] Xilinx Inc
http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf